



US 20060026346A1

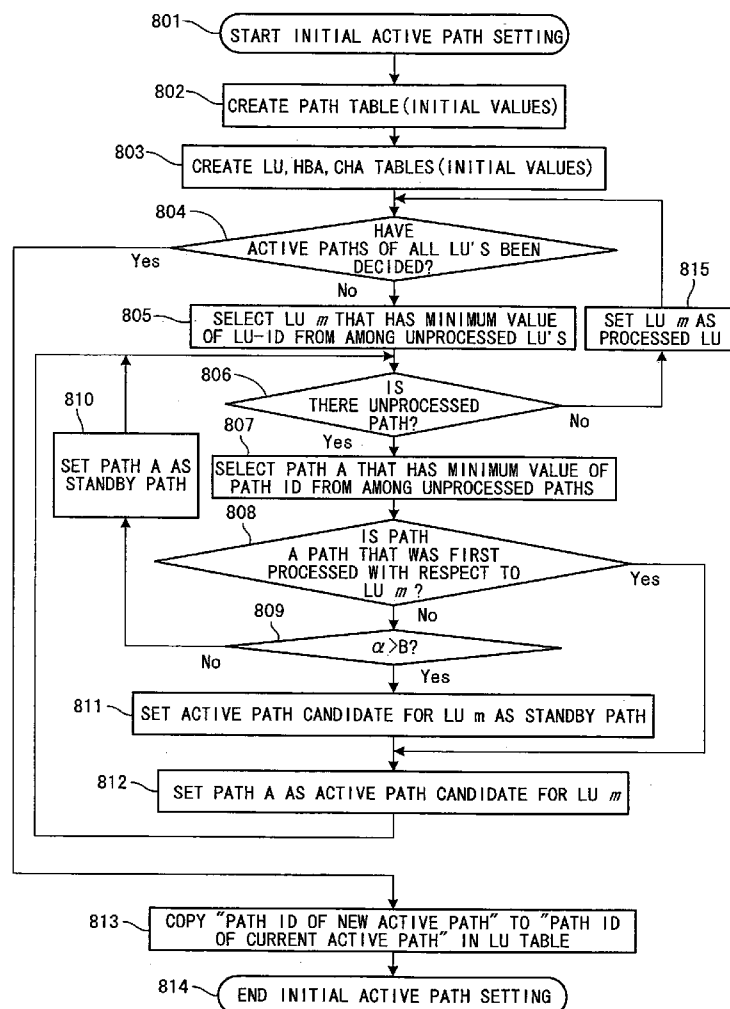
(19) **United States**(12) **Patent Application Publication****Kadoiri et al.**(10) **Pub. No.: US 2006/0026346 A1**(43) **Pub. Date:****Feb. 2, 2006**(54) **COMPUTER SYSTEM FOR LOAD
BALANCE, PROGRAM AND METHOD FOR
SETTING PATHS****Publication Classification**(51) **Int. Cl.**
G06F 12/00 (2006.01)(52) **U.S. Cl.** **711/114; 711/170**(76) Inventors: **Satoshi Kadoiri**, Chigasaki (JP);
Hiroshi Morishima, Yokohama (JP);
Makoto Aoki, Yokohama (JP); **Isao**
Nagase, Ebina (JP)(57) **ABSTRACT**

Provided is a high performance storage system, in which a cache memory is effectively used and access loads are balanced. In a computer system including a computer and a storage system coupled with the computer, the storage system includes one or more channel adapters which communicate with the computer, and a plurality of logical units which store data. The computer includes one or more host bus adapters which communicate with the storage system, and a path management module which manages a path along which the computer accesses the logical unit. The path management module includes a load measurement module which measures an access load on each logical unit, and an active path setting module which sets one or more active paths, along which an access from the computer passes, for each of the logical unit based on the measured access load.

Correspondence Address:

**MATTINGLY, STANGER, MALUR &
BRUNDIDGE, P.C.**
1800 DIAGONAL ROAD
SUITE 370
ALEXANDRIA, VA 22314 (US)
(21) Appl. No.: **10/975,447**(22) Filed: **Oct. 29, 2004**(30) **Foreign Application Priority Data**

Jul. 28, 2004 (JP) 2004-220241



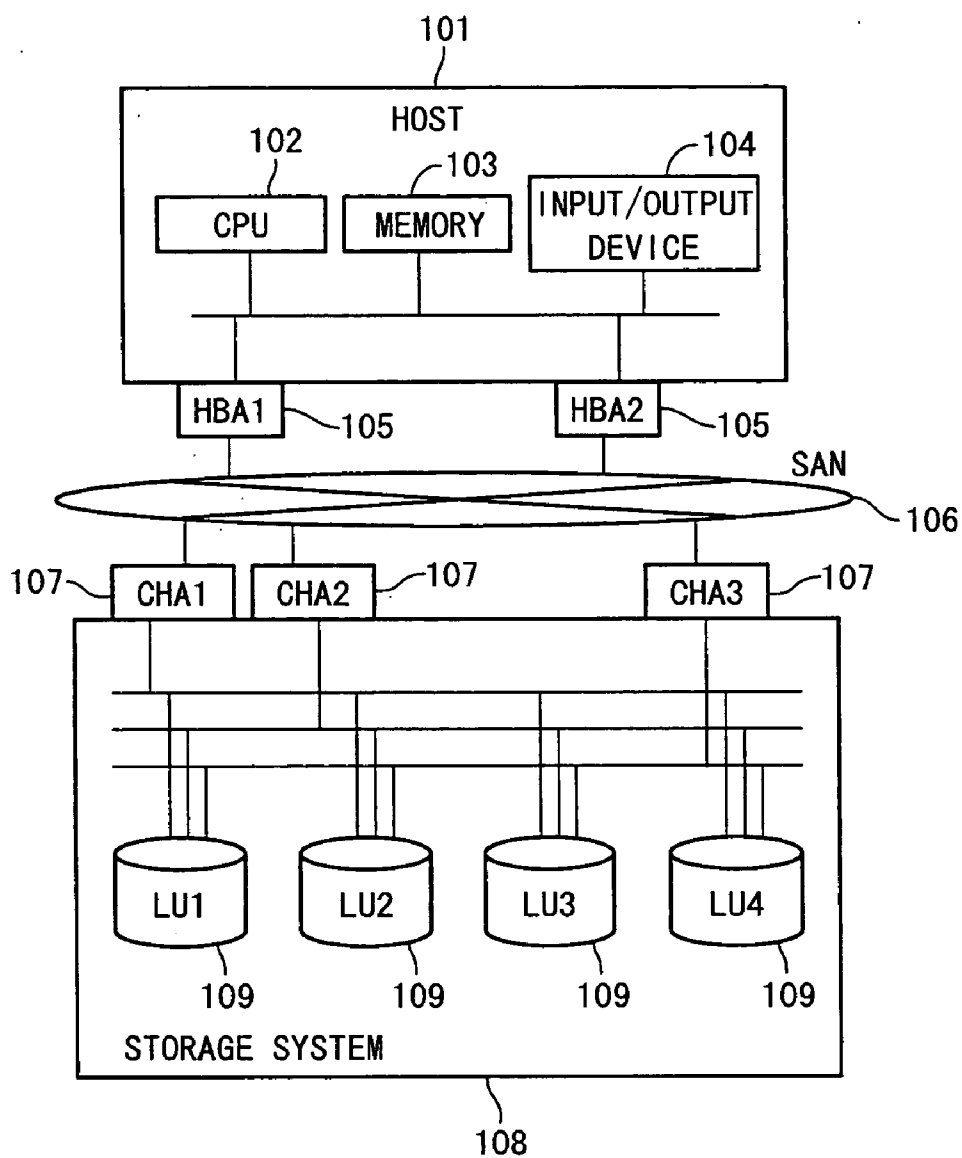


FIG. 1

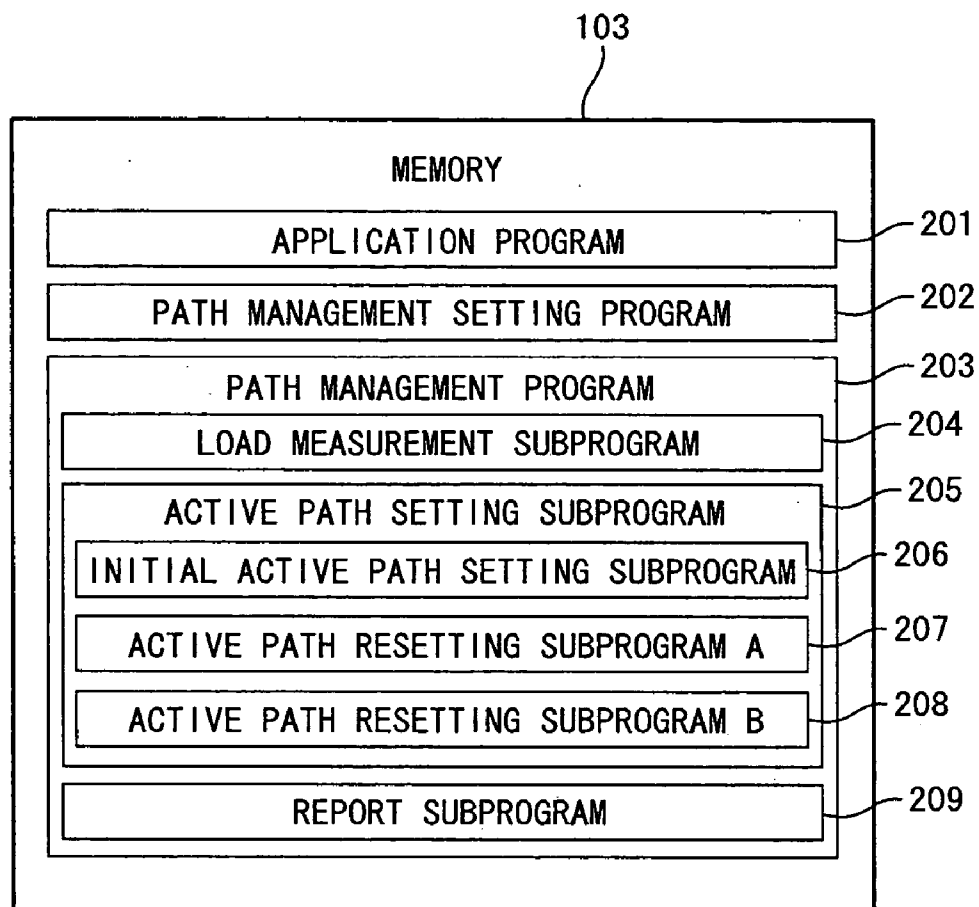


FIG. 2

PATH TABLE

301 PATH NUMBER	302 FAILURE STATE	303 ACTIVE INFORMATION	304 HBA-ID	305 CHA-ID	306 LU-ID	307 MEASURED LOAD	308 ACTIVE PATH SET LU NUMBER
1	No	Yes	1	1	1	50	1
2	No	No	1	2	1	0	0
3	No	No	1	3	1	0	0
4	No	No	2	1	1	0	0
5	No	No	2	2	1	0	0
6	No	No	2	3	1	0	0
7	No	No	1	1	2	0	0
8	No	No	1	2	2	0	0
9	No	No	1	3	2	0	0
10	No	No	2	1	2	0	0
11	No	Yes	2	2	2	20	1
12	No	No	2	3	2	0	0
13	No	No	1	1	3	0	0
14	No	No	1	2	3	0	0
15	No	Yes	1	3	3	10	1
16	No	No	2	1	3	0	0
17	No	No	2	2	3	0	0
18	No	No	2	3	3	0	0
19	No	No	1	1	4	0	0
20	No	No	1	2	4	0	0
21	No	No	1	3	4	0	0
22	No	Yes	2	1	4	60	1
23	No	No	2	2	4	0	0
24	No	No	2	3	4	0	0

FIG. 3

LU TABLE

401 LU-ID	402 MEASURED LOAD	403 PATH ID OF CURRENT ACTIVE PATH	404 PATH ID OF NEW ACTIVE PATH
1	50	111	-1
2	20	222	-1
3	10	133	-1
4	60	214	-1

FIG. 4

HBA TABLE

501 HBA-ID	502 MEASURED LOAD (TOTAL OF OVERALL PATHS)	503 ALLOCATED LOAD	504 ACTIVE PATH SET LU NUMBER
1	60	0	2
2	80	0	2

FIG. 5

CHA TABLE

601 CHA-ID	602 MEASURED LOAD (TOTAL OF OVERALL PATHS)	603 ALLOCATED LOAD	604 ACTIVE PATH SET LU NUMBER
1	110	0	2
2	20	0	1
3	10	0	1

FIG. 6

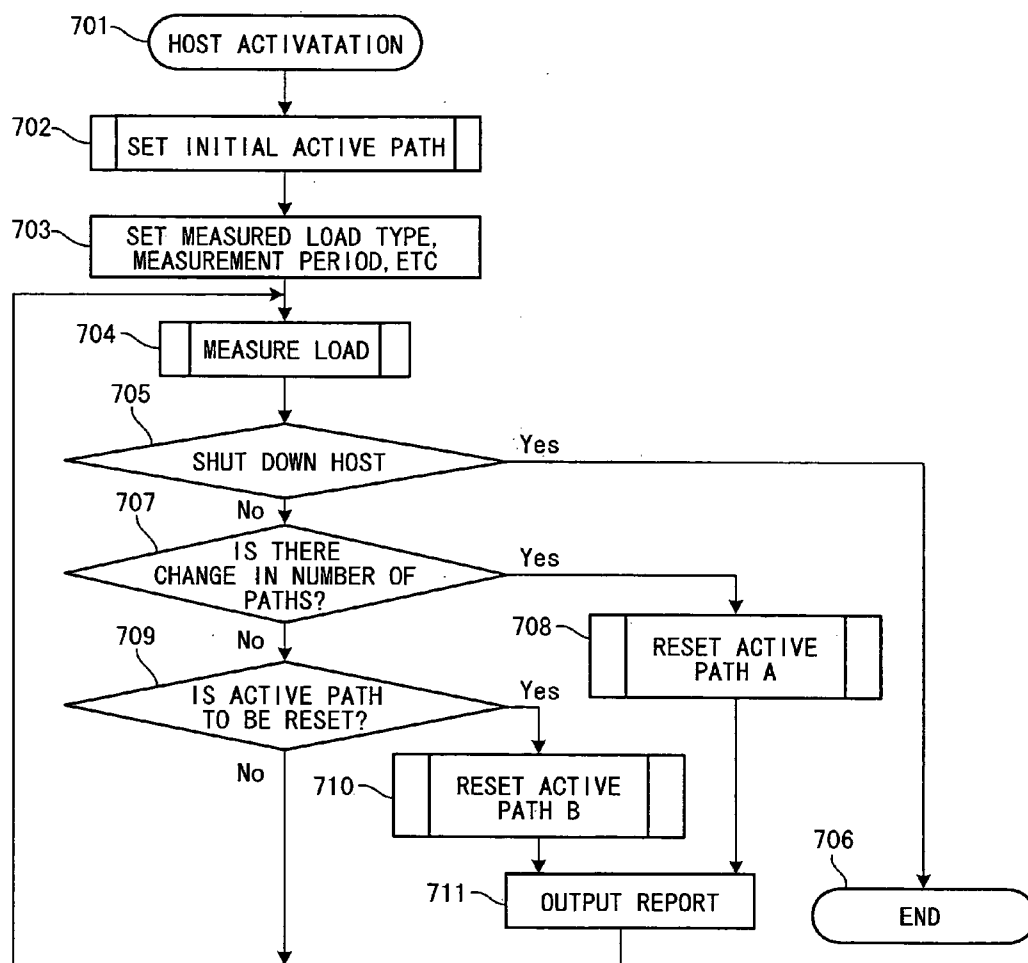


FIG. 7

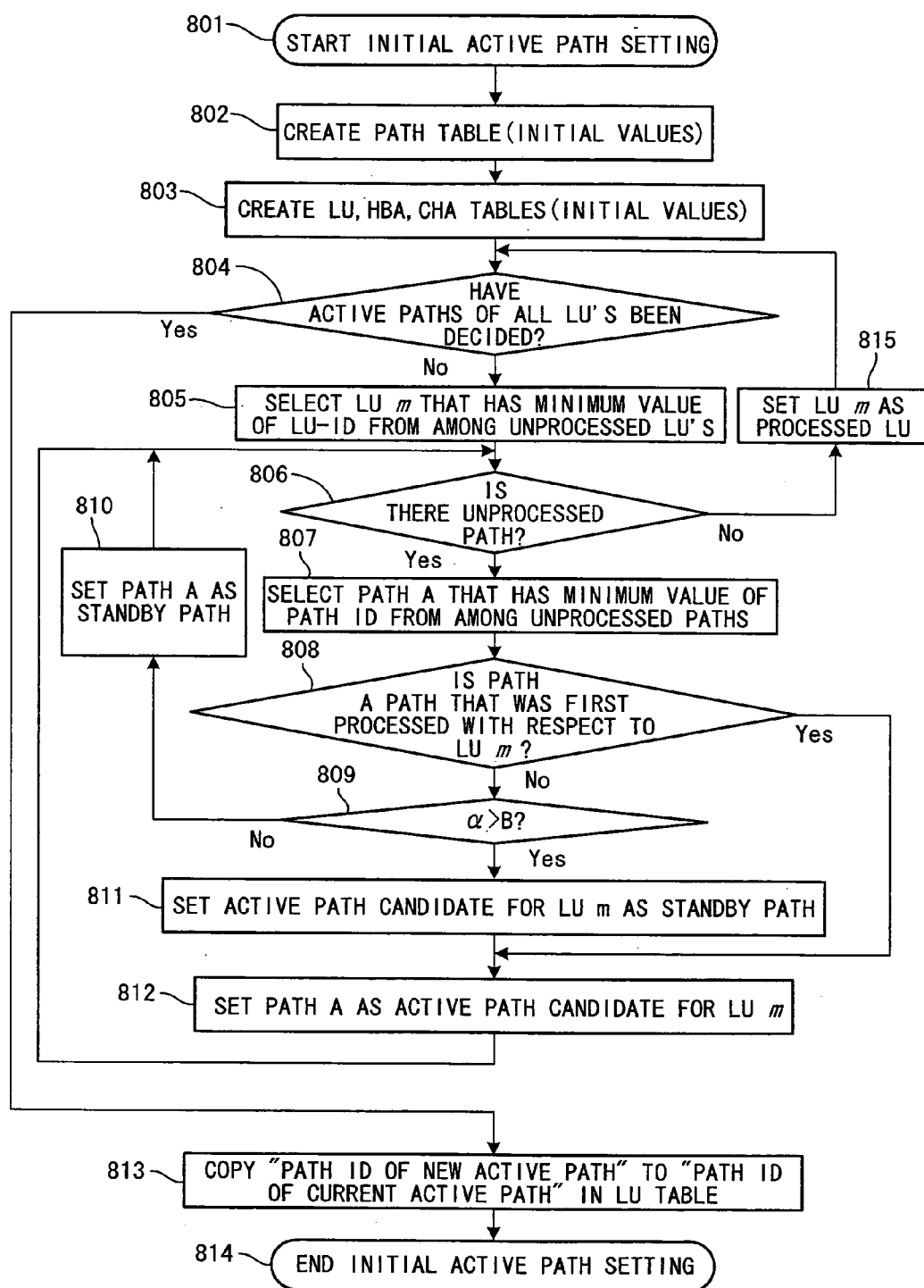


FIG. 8

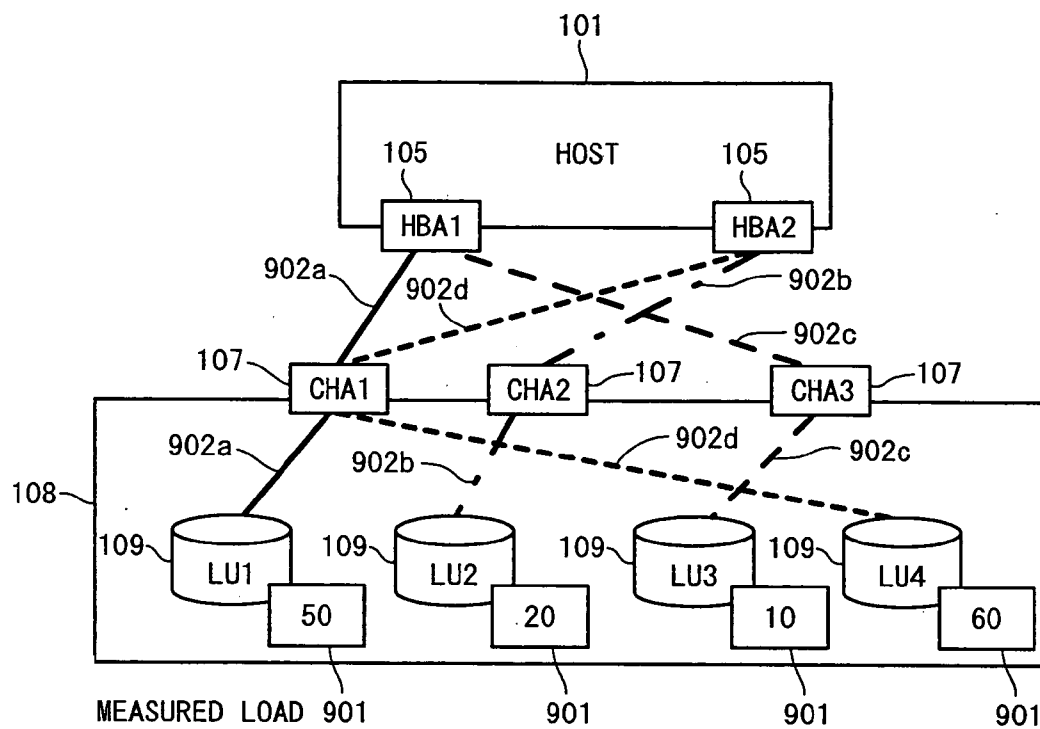


FIG. 9

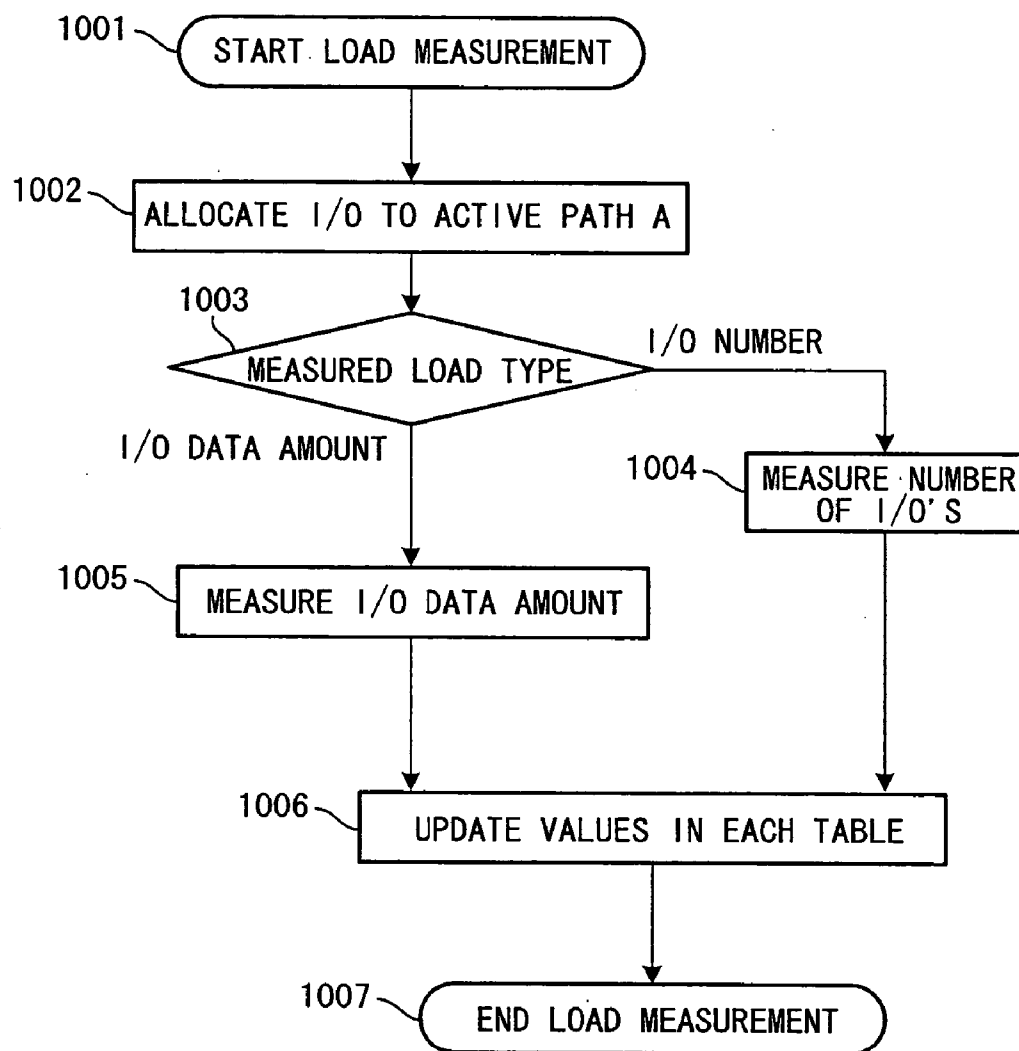


FIG. 10

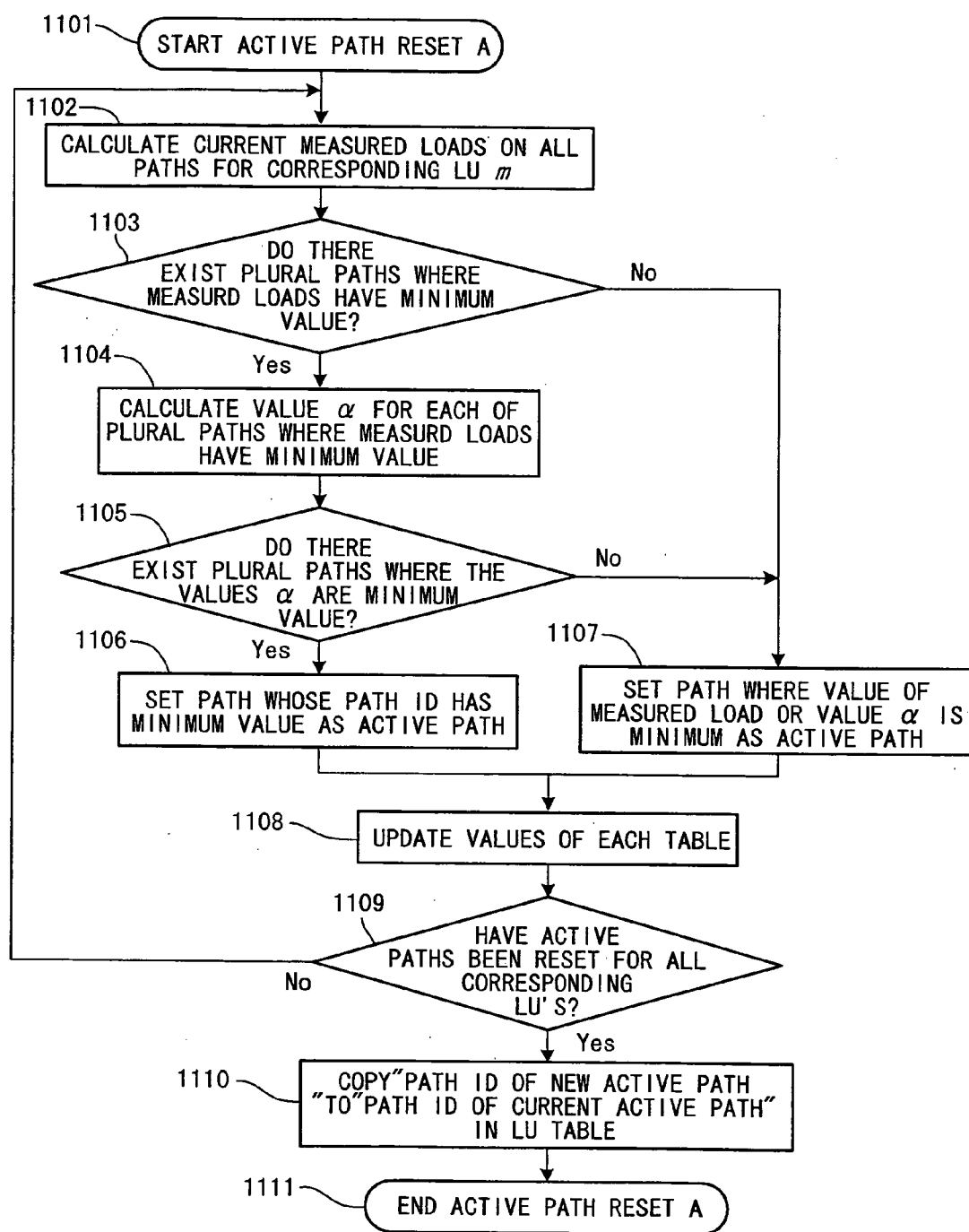


FIG. 11

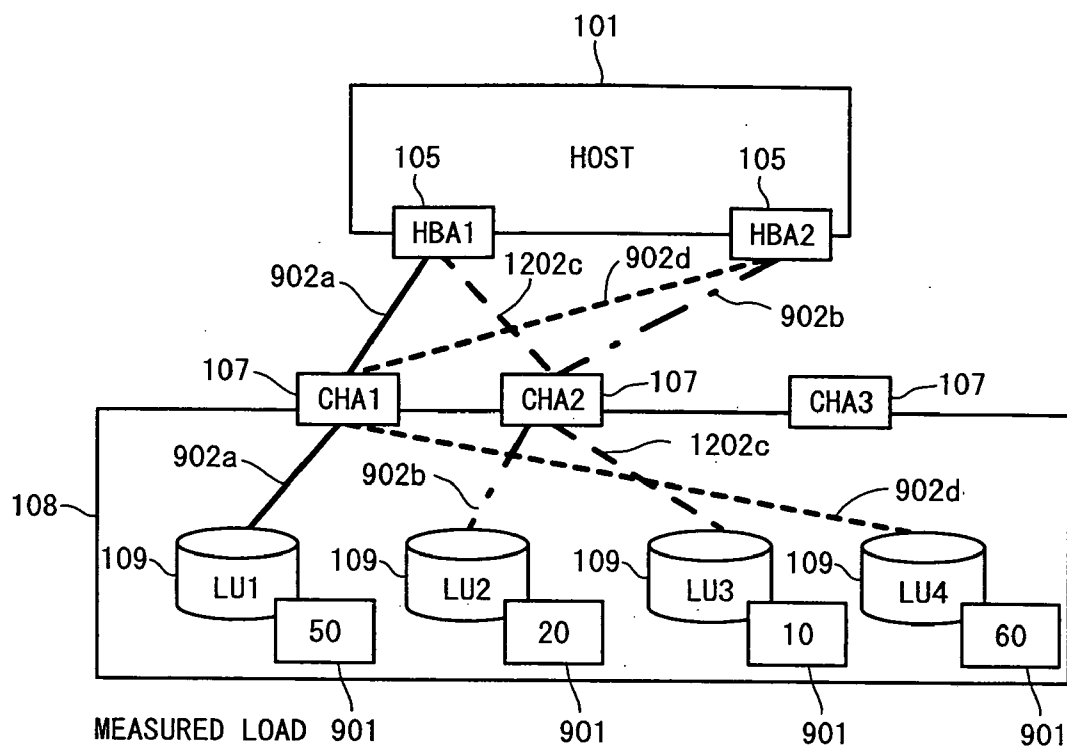


FIG. 12

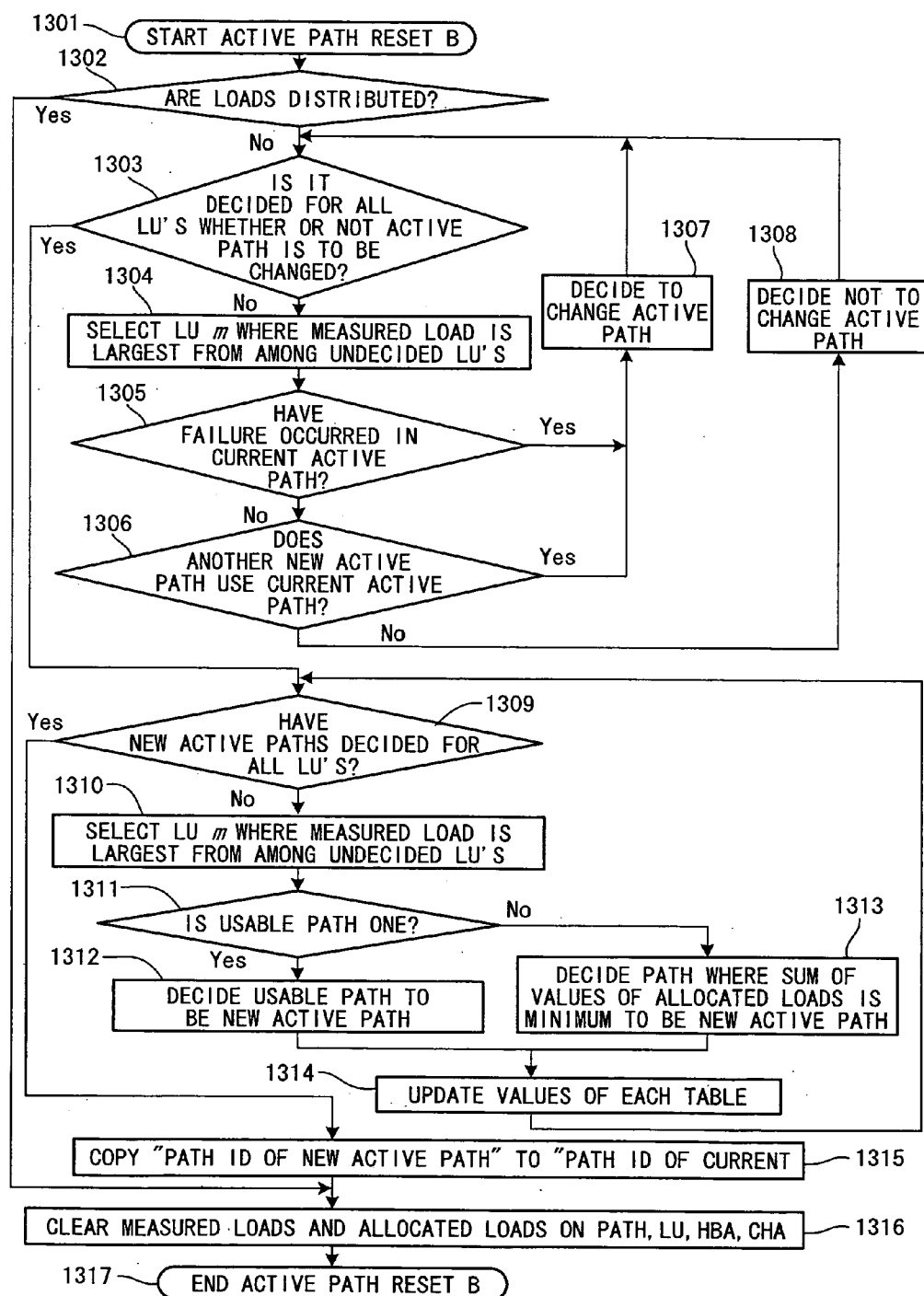


FIG. 13

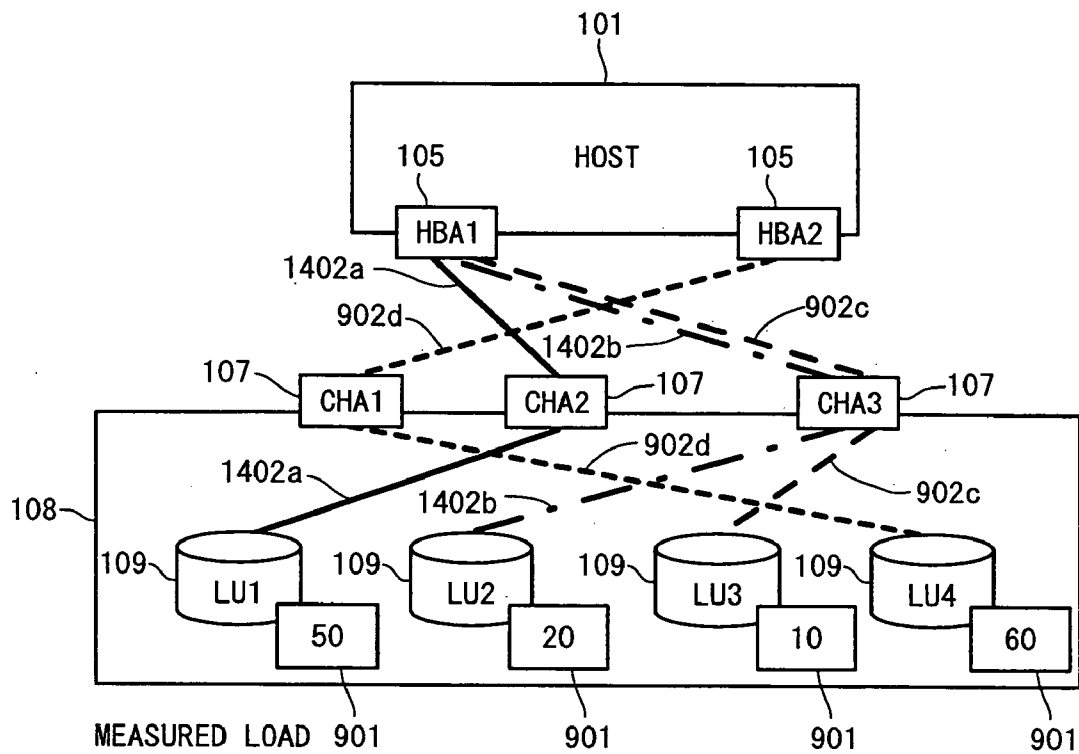


FIG. 14

COMPUTER SYSTEM FOR LOAD BALANCE, PROGRAM AND METHOD FOR SETTING PATHS

CLAIM OF PRIORITY

[0001] The present application claims priority from Japanese application P2004-220241 filed on Jul. 28, 2004, the content of which is hereby incorporated by reference into this application.

BACKGROUND OF THE INVENTION

[0002] This invention relates to an access load distribution for a computer system.

[0003] Examples of a background art include, as disclosed in JP 2000-330924 A, a technique in which a plurality of accessible paths are provided between a storage system and a computer, and accesses made by the computer are distributed across those paths, thereby improving a system performance. According to such a technique for a load balance, access (I/O) loads are distributed across the plurality of paths, thereby preventing the performance from decreasing due to accesses concentrated on a specific path. In addition, when a failure occurs in one path, the accesses are allocated to another normal path so as to allow continuous accesses.

[0004] An alternative technique is proposed in, for example, JP 2002-182978 A, in which a cache memory mounted to a storage system is used for improving a system performance. Examples of the technique include a cache pre-reading technique in which, in addition to data of a block relating to a read request, data of the subsequent block is also read out and stored in the cache memory, thereby improving an access performance.

BRIEF SUMMARY OF THE INVENTION

[0005] Conventional techniques for a load balance are not premised on use of a function using a cache memory of a storage system, such as a data pre-reading function. Therefore, accesses to the same logical unit (LU) are distributed to a plurality of paths that do not allow use of the same cache memory, reducing a cache hit rate. As a result, the function such as a cache pre-reading function is not fully utilized, which leads to a problem in that an increase in performance is suppressed.

[0006] According to an embodiment of this invention, there is provided a computer system, including: a computer; and a storage system coupled with the computer, the storage system including: one or more channel adapters which communicate with the computer; and a plurality of logical units which store data, the computer including: one or more host bus adapters which communicate with the storage system; and a path management module which manages a path along which the computer accesses the logical unit, the path management module including: a load measurement module which measures an access load on each logical unit; and an active path setting module which selects one of: one channel adapter and one or more host bus adapters; and one or more channel adapters and one host bus adapter, along which an access from the computer passes, for each of the logical unit based on the measured access load, and sets an active path that passes the selected channel adapter and the selected host bus adapter.

[0007] According to this invention, access loads are distributed, whereby a system can be prevented from decreasing

in performance due to load concentration. Further, when the same logical unit LU is accessed, the same cache memory can be used, whereby the system improves in performance by pre-reading data and other functions.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] FIG. 1 is a block diagram showing a configuration of a computer system according to an embodiment of this invention.

[0009] FIG. 2 is an explanatory diagram of programs stored in a memory according to the embodiment of this invention.

[0010] FIG. 3 is an explanatory diagram of a path table according to the embodiment of this invention.

[0011] FIG. 4 is an explanatory diagram of an LU table according to the embodiment of this invention.

[0012] FIG. 5 is an explanatory diagram of an HBA table according to the embodiment of this invention.

[0013] FIG. 6 is an explanatory diagram of a CHA table according to the embodiment of this invention.

[0014] FIG. 7 is a flowchart of an path management program according to the embodiment of this invention.

[0015] FIG. 8 is a flowchart of an initial active path setting subprogram according to the embodiment of this invention.

[0016] FIG. 9 is a block diagram showing active paths that are set according to the embodiment of this invention.

[0017] FIG. 10 is a flowchart of a load measurement subprogram according to the embodiment of this invention.

[0018] FIG. 11 is a flowchart of an active path resetting subprogram A according to the embodiment of this invention.

[0019] FIG. 12 is a block diagram showing an example of the active paths that are reset according to the embodiment of this invention.

[0020] FIG. 13 is a flowchart of an active path resetting subprogram B according to the embodiment of this invention.

[0021] FIG. 14 is a block diagram showing another example of the active paths that are reset according to the embodiment of this invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0022] Hereinafter, description will be made of embodiments of this invention with reference to the drawings.

[0023] FIG. 1 is a block diagram showing a configuration of a computer system according to the embodiment of this invention.

[0024] The computer system according to this invention includes a host **101**, a storage system **108**, and a storage area network (SAN) **106** which communicably connects the host **101** and the storage system **108** with each other.

[0025] The host **101** is a computer, which writes data to the storage system **108** or reads out data from the storage

system 108. The host 101 includes a CPU 102, a memory 103, an input/output device 104, and a host bus adapter (HBA) 105.

[0026] The CPU 102 is a processor which executes a program stored in the memory 103.

[0027] The memory 103 is, for example, a semiconductor memory, which stores programs which are shown in FIG. 2 and are to be executed by the CPU 102. The memory 103 further stores tables shown in FIGS. 4 through 7.

[0028] The input/output device 104 includes, for example, a display screen and a keyboard (not shown), and may further include other input/output means such as a mouse and a printer.

[0029] One or plural HBAs 105 are provided to the host 101, allowing communication with the storage system 108 through the SAN 106. In the example of FIG. 1, two HBAs 105 are provided to the host 101. Here, an HBA 1 and an HBA 2 are the HBAs 105 which have HBA identifiers (HBA-IDs) of "1" and "2", respectively. The host 101 can use either of the HBAs 105 to communicate with the storage system 108.

[0030] The storage system 108 is, for example, a disk array system including a plurality of disk drives and a controller which controls the disk drives (not shown). In the storage system 108, one or more logical units (LUs) 109 are set.

[0031] The LU 109 is a unit which can be recognized by the host 101 as a single disk drive. The LU 109 is logically a single disk drive, but may physically be a partial region of a single disk drive or a partial or entire region of a plurality of disk drives. In the example of FIG. 1, four LUs 109 are set in the storage system 108. Here, LUs 1 through 4 are the LUs 109 which have LU identifiers (LU-IDs) of "1" through "4", respectively.

[0032] The storage system 108 further includes one or more channel adapters (CHAs) 107. The CHA 107 communicates with the host 101 through the SAN 106. In the example of FIG. 1, three CHAs 107 are provided to the storage system 108. Here, CHAs 1 through 3 are the CHAs 107 which have CHA identifiers (CHA-IDs) of "1" through "3", respectively. The storage system 108 can use any of the CHAs 107 to communicate with the host 101.

[0033] The CHA 107 normally includes a cache memory (not shown). When the CHA 107 receives a read request for data from the host 101 and when the data of the read request exists in the cache memory, the CHA 107 does not access the LU 109 and returns the data in the cache memory to the host 101. As a result, an access to the LU 109 (in other words, access to a disk drive forming the LU 109) is omitted, improving an access speed.

[0034] A cache memory having the same function may be provided to the HBA 105.

[0035] The SAN 106 communicably connects the HBA 105 of the host 101 and the CHA 107 of the storage system 108 with each other. Each HBA 105 can communicate with any of the CHAs 107. Similarly, each CHA 107 can communicate with any of the HBAs 105. The SAN 106 typically includes a Fibre Channel switch and cables.

[0036] The term "path" used herein represents a data access route extending from the host 101 to the LU 109. In the example of FIG. 1, there are two HBAs 105 and three CHAs 107, so there exist six paths that extend from the host 101 to one of the LUs 109. The path is identified by a path ID. The path ID is defined based on the identifiers of the HBA 105, the CHA 107, and the LU 109 which exist along the corresponding path (HBA-ID, CHA-ID, and LU-ID, respectively). For example, the path extending through the HBA 1 and the CHA 3 to the LU 2 has a path ID of "132".

[0037] FIG. 2 is an explanatory diagram of programs stored in the memory 103 according to the embodiment of this invention.

[0038] The memory 103 stores an application program 201, a path management setting program 202, and a path management program 203. The programs are all executed by the CPU 102.

[0039] The path management setting program 202 is a program for setting parameters and the like of the path management program 203.

[0040] The path management program 203 is a program for realizing a load balance according to this invention. The path management program 203 sets an access path with respect to each LU 109 according to an algorithm described later. A data access (I/O) requested by the application program 201 is made via the access path set by the path management program 203.

[0041] The path management program 203 includes a load measurement subprogram 204, an active path setting subprogram 205, and a report subprogram 209.

[0042] The report subprogram 209 outputs information on active path settings, information on states of a load on each path, and other such information to the input/output device 104. The report subprogram 209 outputs those pieces of information at least when an active path is reset.

[0043] The active path will be described later with reference to FIG. 3.

[0044] Further, the active path setting subprogram 205 includes an initial active path setting subprogram 206, an active path resetting subprogram A 207, and an active path resetting subprogram B 208.

[0045] Those subprograms will be described in detail later.

[0046] FIG. 3 is an explanatory diagram of a path table according to the embodiment of this invention.

[0047] A path table 300 is a table stored in the memory 103 and includes information indicating the states of all the paths existing in the computer system of FIG. 1 (in other words, the host 101 and the storage system 108 connected thereto). The path table 300 is created by the initial active path setting subprogram 206 and updated by the load measurement subprogram 204, the active path resetting subprogram A 207, and the active path resetting subprogram B 208.

[0048] In the path table 300, a path number 301 represents the number of a path, which is uniquely assigned to every path.

[0049] A failure state 302 indicates whether a failure has occurred in the path. When no failures have occurred in the

HBA **105**, the CHA **107**, and the LU **109**, which form a path, the failure state **302** has a value of “No”. In contrast, when a failure has occurred in either of those components, the failure state **302** has a value of “Yes”.

[0050] An active information **303** indicates whether the path is set as an active path. When the path is set as an active path, the active information **303** has a value of “Yes”. When the path is not set as an active path, the active information **303** has a value of “No”.

[0051] The term “active path” used herein represents a path along which an access (I/O) is made from the host **101** to the LU **109**. An inactive path is a standby path that does not contribute to the data access. For example, FIG. 3 shows that there are six accessible paths whose path numbers **301** are “1” through “6” between the host **101** and the LU **1**. However, the host **101** actually accesses the LU **1** via only the active path (in the example of FIG. 3, the path whose path number **301** is “1”).

[0052] The active path is first created by the initial active path setting subprogram **206**. After that, as the need arises (for example, when a failure occurs in the active path, or when loads are concentrated on a specific component), the active path resetting subprogram A **207** or the active path resetting subprogram B **208** sets (resets) one of standby paths as a new active path, and further sets the original active path as a standby path. Setting and resetting processes for an active path will be described in detail later.

[0053] An HBA-ID **304** represents the identifier (HBA-ID) of the HBA **105** existing along the path.

[0054] A CHA-ID **305** represents the identifier (CHA-ID) of the CHA **107** existing along the path.

[0055] An LU-ID **306** represents the identifier (LU-ID) of the LU **109** existing along the path.

[0056] A measured load **307** indicates an amount of load which is actually measured for each path. The amount of load is represented by the number of accesses (I/Os) or the amount of access data per predetermined time period, which is measured by the load measurement subprogram **204** as shown in FIG. 10.

[0057] An active path set LU number **308** indicates the number of LUs for which the path is set as an active path. The active path set LU number **308** of the path whose active information **303** is “Yes” is “1”, and the active path set LU number **308** of the path whose active information **303** is “No” is “0”.

[0058] A setting process for the path table **300** will be described in detail later.

[0059] FIG. 4 is an explanatory diagram of an LU table according to the embodiment of this invention.

[0060] An LU table **400** is a table stored in the memory **103** and includes information indicating the states of all the LUs **109** existing in the storage system **108** connected to the host **101** of FIG. 1. The LU table **400** is created by the initial active path setting subprogram **206** and updated by the load measurement subprogram **204**, the active path resetting subprogram A **207**, and the active path resetting subprogram B **208**.

[0061] In the LU table **400**, an LU-ID **401** represents the LU-ID identifying each LU. In the example of FIG. 4, a value of the LU-ID **401** varies from “1” through “4”.

[0062] A measured load **402** indicates an amount of load which is actually measured for each LU. A value of the measured load **402** on each LU is obtained by summing up the values of the measured load **307** of the path table **300** in terms of the corresponding LU-ID **306**. For example, the measured load **402** on the LU **109** whose LU-ID **401** is “1” (in other words, LU **1**) is represented by the total value of the measured load **307** on the paths whose LU-ID **306** is “1” in the path table **300**. In the example of FIGS. 3 and 4, the measured load **402** on the LU **1** is “50”.

[0063] A path ID of current active path **403** indicates the path ID of a path that is currently set as an active path for each LU. In other words, the path ID of current active path **403** corresponds to the path ID of the path whose active path set LU number **308** is “1” in the path table **300**.

[0064] A path ID of new active path **404** indicates the path ID of a path that is to be subsequently set as an active path for each LU. When the path that is to be subsequently set as an active path is undecided, the path ID of new active path **404** has a value of “-1”.

[0065] A setting process for the LU table **400** will be described in detail later.

[0066] FIG. 5 is an explanatory diagram of an HBA table according to the embodiment of this invention.

[0067] An HBA table **500** is a table stored in the memory **103** and includes information indicating the states of all the HBAs **105** existing in the host **101** of FIG. 1. The HBA table **500** is created by the initial active path setting subprogram **206** and updated by the load measurement subprogram **204**, the active path resetting subprogram A **207**, and the active path resetting subprogram B **208**.

[0068] In the HBA table **500**, an HBA-ID **501** represents the HBA-ID identifying each HBA. In the example of FIG. 5, a value of the HBA-ID **501** varies between “1” and “2”.

[0069] A measured load **502** indicates an amount of load which is actually measured for each HBA. A value of the measured load **502** on each LU is obtained by summing up the values of the measured load **307** of the path table **300** in terms of the corresponding HBA-ID **304**. For example, the measured load **502** on the HBA **105** whose HBA-ID **501** is “1” (in other words, HBA **1**) is represented by the total value of the measured load **307** for the paths whose HBA-ID **304** is “1” in the path table **300**. In the example of FIGS. 3 and 5, the measured load **502** of the HBA **1** is “60”.

[0070] The field of an allocated load **503** has a value of loads on active paths that have already been allocated to each HBA **105**, which is stored when the active path is reset.

[0071] An active path set LU number **504** indicates the number of LUs **109** for which each HBA **105** is set as an active path. The active path set LU number **504** of each HBA **105** is obtained by summing up the values of the active path set LU numbers **308** of the path table **300** in terms of the corresponding HBA-ID **304**. For example, the active path set LU number **504** of the HBA **105** whose HBA-ID **501** is “1” (in other words, HBA **1**) is represented by the total value of the active path set LU numbers **308** for the paths whose

HBA-ID **304** is “1” in the path table **300**. In the example of **FIGS. 3 and 5**, the active path set LU number **504** of the HBA **1** is “2”.

[**0072**] A setting process for the HBA table **500** will be described in detail later.

[**0073**] **FIG. 6** is an explanatory diagram of a CHA table according to the embodiment of this invention.

[**0074**] A CHA table **600** is a table stored in the memory **103** and includes information indicating the states of all the CHAs **107** existing in the storage system **108** connected to the host **101** of **FIG. 1**. The CHA table **600** is created by the initial active path setting subprogram **206** and updated by the load measurement subprogram **204**, the active path resetting subprogram A **207**, and the active path resetting subprogram B **208**.

[**0075**] In the CHA table **600**, a CHA-ID **601** represents the CHA-ID identifying each CHA. In the example of **FIG. 6**, a value of the CHA-ID **601** varies from “1” through “3”.

[**0076**] A measured load **602** indicates an amount of load which is actually measured for each CHA. A value of the measured load **602** on each CHA is obtained by summing up the values of the measured load **307** of the path table **300** in terms of the corresponding CHA-ID **305**. For example, the measured load **602** on the CHA **107** whose CHA-ID **601** is “1” (in other words, CHA **1**) is represented by the total value of the measured load **307** on the paths whose CHA-ID **305** is “1” in the path table **300**. In the example of **FIGS. 3 and 6**, the measured load **602** of the CHA **1** is “110”.

[**0077**] The field of an allocated load **603** has a value of loads on active paths that have already been allocated to each CHA **107**, which is stored when the active path is reset.

[**0078**] An active path set LU number **604** indicates the number of LUs **109** for which each CHA **107** is set as an active path. The active path set LU number **604** of each CHA **107** is obtained by summing up the values of the active path set LU numbers **308** of the path table **300** in terms of the corresponding CHA-ID **305**. For example, the active path set LU number **604** of the CHA **107** whose CHA-ID **601** is “1” (in other words, CHA **1**) is represented by the total value of the active path set LU numbers **308** for the paths whose CHA-ID **305** is “1” in the path table **300**. In the example of **FIGS. 3 and 6**, the active path set LU number **604** of the CHA **1** is “2”.

[**0079**] A setting process for the CHA table **600** will be described in detail later.

[**0080**] The LU table **400**, the HBA table **500**, and the CHA table **600** are created by the following method.

[**0081**] For example, the HBA **105** of the host **101** may detect the LU **109**, and the operating system (OS) of the host **101** may obtain information on the detected LU **109** from the HBA **105**. The host **101** can obtain information on a path (for example, information indicating which HBA **105** and which CHA **107** exist on the path) by using a SCSI command with respect to the detected LU **109**. Based on the information, the host **101** grasps the usable LU **109**, HBA **105**, and CHA **107**, and can create the LU table **400**, the HBA table **500**, and the CHA table **600**. Accordingly, the host **101** can detect the usable LU **109** and CHA **107**.

[**0082**] When a plurality of hosts **101** can detect the same LU **109** and CHA **107**, those plurality of hosts **101** share the same LU **109** and CHA **107** that have been detected.

[**0083**] In contrast, the LU **109** that cannot be detected by the OS of the host **101** can also be set by a function such as zoning performed by a Fibre Channel switch (not shown) of the SAN **106** or LUN masking performed by the storage system **108**. Therefore, the function allows such a setting that the plurality of hosts **101** respectively use different LUs **109**.

[**0084**] **FIG. 7** is a flowchart of the path management program **203** according to the embodiment of this invention.

[**0085**] The host **101** is activated to start execution of the path management program **203** (**701**), and an initial active path is first set (**702**). To be specific, as shown in **FIGS. 8 and 9**, the initial active path setting subprogram **206** called by the path management program **203** sets the initial active path.

[**0086**] Next, a measured load type and a measurement period are set (**703**). The measured load type represents an object to be measured by the load measurement subprogram **204**, and is set by being selected between an “I/O number” and an “I/O data amount” by the user of the host **101**. The measurement period represents a period during which the load measurement subprogram **204** measures the object to be measured, and the user of the host **101** sets an arbitrary period as the measurement period. The setting is executed by the path management setting program **202**.

[**0087**] Next, the load is measured according to the settings made in the step **703** (**704**). To be specific, the load measurement subprogram **204** called by the path management program **203** measures the load according to the settings made in the step **703** (see **FIG. 10**). Every time an I/O (in other words, data access) occurs, the load measurement subprogram **204** measures the load.

[**0088**] Next, it is judged whether the host **101** is to be shut down (**705**). To be specific, it is judged whether the user has issued a command to shut down the host **101**. When it is judged that the host **101** is to be shut down, the path management program **203** ends (**706**).

[**0089**] On the other hand, when it is judged that the host **101** is not to be shut down, it is then judged whether there is a change in the number of paths (**707**). Here, the case where there is a change in the number of paths includes: a case where a failure occurs in the HBA **105**, CHA **107**, or LU **109**, disabling the use of one or more paths; a case where a recovery is made from the failure that occurred in the HBA **105**, CHA **107**, or LU **109**; and a case where another HBA **105**, CHA **107**, or LU **109** is added. When there is a change in the number of paths, the active path is reset (**708**). To be specific, the active path resetting subprogram A **207** called by the path management program **203** resets the active path (see **FIGS. 11 and 12**).

[**0090**] On the other hand, when there is no change in the number of paths, it is judged whether the timing for resetting the active path has been reached (**709**). The user of the host **101** can preset an interval (for example, 1 hour, 1 day, 1 week, 1 month, or 1 year) for resetting an active path. When the user sets “1 hour”, it is judged in the step **709** whether or not 1 hour has elapsed since the active path was reset last

time (or was initially set). When 1 hour has elapsed, the timing for resetting the active path has been reached, so the active path is reset (710). To be specific, the active path resetting subprogram B 208 called by the path management program 203 resets the active path (see FIGS. 13 to 16).

[0091] It should be noted that the resetting of the active path is not only executed at predetermined intervals as described above, but also may be executed in response to a request made by the user of the host 101. In this case, it is judged in the step 709 whether the user has issued an active path resetting request for the active path. When the active path resetting request has been issued, the timing for resetting the active path has been reached, so the procedure advances to the step 710.

[0092] On the other hand, when the above-mentioned predetermined interval has not elapsed, and when the active path resetting request has not been issued from the user, the timing for resetting the active path has not been reached yet, so the procedure returns to the step 704.

[0093] Meanwhile, after the active path resetting subprogram A 207 or the active path resetting subprogram B 208 is executed, a report is outputted (711). To be specific, the report subprogram 209 displays information on the reset active path on the input/output device 104 of the host 101. After the report is outputted, the procedure returns to the step 704.

[0094] FIG. 8 is a flowchart of the initial active path setting subprogram 206 according to the embodiment of this invention.

[0095] By being called in the step 702 of FIG. 7, the execution of the initial active path setting subprogram 206 starts (801). At this time, the path management program 203 recognizes all the LUs 109 (in this embodiment, LU 1 to LU 4) that can be accessed by the host 101.

[0096] First, the initial active path setting subprogram 206 creates the path table 300 (802). At this time, values in the path table 300 are all initial values. In other words, the values of the failure state 302 and active information 303 are all set to "No", and the values of the measured load 307 and active path set LU number 308 are all set to "0".

[0097] Next, the initial active path setting subprogram 206 creates the LU table 400, the HBA table 500, and the CHA table 600 (803). At this time, values in each table are all initial values. In other words, the values of the measured loads 402, 502, and 602, allocated loads 503 and 603, and active path set LU numbers 504 and 604 are all set to "0". The values of the path ID of current active path 403 and path ID of new active path 404 are undecided.

[0098] Next, it is judged whether the active paths for all the LUs 109 have been decided (804).

[0099] When the active paths for all the LUs 109 have not been decided, the LU 109 (LU m, where "m" represents an LU-ID) that has the minimum value of LU-ID is selected from among unprocessed LUs 109 (805).

[0100] Next, it is judged whether there is an unprocessed path among the paths for the LU m (806). When there is no unprocessed path, an active path candidate for the LU m at this time is decided as the active path, so the LU m is set as

a processed LU (815). Then, in order to decide the active path for the subsequent LU 109, the procedure returns to the step 804.

[0101] Meanwhile, when there is left an unprocessed path, the active path for the LU m is undecided. Thus, in order to decide the active path, a path A that has the minimum value of path ID is selected from among unprocessed paths for the LU m (807).

[0102] Next, it is judged whether the path A is the path that was first selected for the LU m (808). When the path A is the path that was first selected, the procedure advances to a step 812 in order to set the path A as the active path candidate for the LU m.

[0103] On the other hand, when the path A is not the path that was first selected, in order to judge whether to set the path A as the active path candidate, it is judged whether a value a is larger than a value B (809). The value a used herein represents the total value of the active path set LU number 504 of the HBA 105 on the active path candidate at that time and the active path set LU number 604 of the CHA 107 on the active path candidate at that time. The value B represents the total value of the active path set LU number 504 of the HBA 105 on the path A and the active path set LU number 604 of the CHA 107 on the path A.

[0104] As a result, when the value a is not larger than the value B, the path A does not become the active path candidate for the LU m. Thus, the path A is set as the standby path (810), and the procedure returns to the step 806. To be specific, in the step 810, the active information 303 corresponding to the path A in the path table 300 is set to "No".

[0105] On the other hand, when the value a is larger than the value B, in order to set the path A as the active path candidate for the LU m, the active path candidate for the LU m at the current time is set as the standby path (811). To be specific, in the path table 300, the active information 303 corresponding to the active path candidate at the current time is set to "No". In addition, in the path table 300, the active path set LU numbers 308 corresponding to the active path candidate at the current time decrements by 1. Further, with regard to the HBA 105 corresponding to the active path candidate at the current time, the active path set LU number 504 in the HBA table 500 decrements by 1. Furthermore, with regard to the CHA 107 corresponding to the active path candidate at the current time, the active path set LU number 604 in the CHA table 600 decrements by 1.

[0106] Next, the path A is set as the active path candidate for the LU m (812). To be specific, in the path table 300, the active information 303 corresponding to the path A is set to "Yes". In addition, in the path table 300, the active path set LU numbers 308 corresponding to the path A increments by 1. Further, with regard to the HBA 105 corresponding to the path A, the active path set LU number 504 in the HBA table 500 increments by 1. Furthermore, with regard to the CHA 107 corresponding to the path A, the active path set LU number 604 in the CHA table 600 increments by 1. Moreover, in the LU table 400, the value of the path ID of new active path 404 corresponding to the LU m is set to the value of the path ID of the path A.

[0107] After that, the procedure returns to the step 806 to process the subsequent path.

[0108] Meanwhile, when the active path for all the LUs 109 is decided in the step 804, in order to put the decided active path into actual operation, the value of the path ID of new active path 404 in the LU table 400 is copied to the path ID of current active path 403 (813). As a result, the path management program 203 starts to operate the new active path. In other words, the path management program 203 allocates the data access (I/O) requested by the application program 201 to the new active path.

[0109] After all those processes, the initial active path setting subprogram 206 ends its execution (814).

[0110] FIG. 9 is a block diagram showing active paths that are set according to the embodiment of this invention.

[0111] FIG. 9 shows an example of results obtained by subjecting the system shown in FIG. 1 to the execution of the initial active path setting subprogram 206 described in FIG. 8.

[0112] In FIG. 9, the path IDs of the set active paths are "111", "222", "333", and "214". In other words, an active path 902a extends to the LU 1 via the HBA 1 and the CHA 1. An active path 902b extends to the LU 2 via the HBA 2 and the CHA 2. An active path 902c extends to the LU 3 via the HBA 1 and the CHA 3. An active path 902d extends to the LU 4 via the HBA 2 and the CHA 1.

[0113] FIG. 9 shows measured loads 901 indicating an example of the load measured by the load measurement subprogram 204. The measured loads 901 will be described in detail with reference to FIG. 10 and the subsequent figures.

[0114] FIG. 10 is a flowchart of the load measurement subprogram 204 according to the embodiment of this invention.

[0115] By being called in the step 704 of FIG. 7, the execution of the load measurement subprogram 204 starts (1001).

[0116] First, the load measurement subprogram 204 allocates the data access (I/O) from the host 101 to any of the LUs 109, to the active path A set for the LU 109 (1002).

[0117] Next, the measured load type is judged (1003). The measured load type is set in the step 703 of FIG. 7.

[0118] When it is judged that the "I/O number" is set as the measured load type, the number of data I/Os issued for the active path A is measured (1004).

[0119] On the other hand, when it is judged that the "I/O data amount" is set as the measured load type, the data amount of the data I/Os issued for the active path A is measured (1005).

[0120] Next, the values in each table are updated (1006). In other words, the value measured in the step 1004 or 1005 is added to the value of the measured load 307 corresponding to the active path A in the path table 300. In addition, in the LU table 400, the HBA table 500, and the CHA table 600, the values measured in the step 1004 or 1005 are added to the values of the measured loads 402, 502, and 602 on the LU, HBA, and CHA, respectively, existing along the active path A.

[0121] After all those processes, the load measurement subprogram 204 ends its execution (1007). FIG. 9 shows the

measured loads 901 indicating an example of the values measured by the load measurement subprogram 204.

[0122] It should be noted that the values in the tables shown in FIGS. 3 to 6 are measured when the measured loads 901 shown in FIG. 9 are measured.

[0123] FIG. 11 is a flowchart of the active path resetting subprogram A 207 according to the embodiment of this invention.

[0124] In the computer system of FIG. 1, in the case where there is a change in the number of paths, the active path resetting subprogram A 207 is called in the step 708 of FIG. 7 to start its execution (1101). Here, the case where there is a change in the number of paths includes: a case where a failure occurs in the HBA 105 or CHA 107 included in the active path; a case where another LU 109 is added; and a case where a recovery is made from the failure that occurred in the HBA 105 or CHA 107.

[0125] First, the active path resetting subprogram A 207 according to the embodiment of this invention calculates the measured loads on all the paths extending to the corresponding LU m (1102). When a failure occurs in the active path in the step 1102, the corresponding LU m represents the LU 109 relating to the active path. Alternatively, when another LU 109 is added, the corresponding LU m represents the added LU 109. Alternatively, when a recovery is made from the failure that occurred in the HBA 105 or CHA 107, the corresponding LU m represents the LU 109 that can be accessed via the HBA 105 or CHA 107. Also in the step 1102, the measured loads on all the paths extending to the corresponding LU m is a value obtained by summing up the values of the measured loads 502 and 602 on the HBAs 105 and CHAs 107 included in the active paths on a path basis. However, the value measured for the active path experiencing a failure is not included.

[0126] Next, the values of the measured loads calculated in the step 1102 are compared with each other, and it is judged whether there exist a plurality of paths where the measured loads have the minimum value (1103). When there do not exist the plurality of paths where the measured loads have the minimum value, the path where the measured load has the minimum value is set as the active path (1107).

[0127] On the other hand, when there exist the plurality of paths where the measured loads calculated in the step 1102 have the minimum value, the value a is calculated for each of the paths (1104). The value a is obtained by summing up the active path set LU numbers 504 and active path set LU numbers 604 in the HBA table 500 and CHA table 600 on a path basis.

[0128] Next, the values a calculated for the paths are compared with each other, it is judged whether there exist a plurality of paths where the values a are the minimum value (1105). When there do not exist the plurality of paths where the measured loads are the minimum value, the path where the measured load has the minimum value is set as the active path (1107).

[0129] On the other hand, when there exist the plurality of paths where the values a are the minimum value, the path whose path ID has the minimum value is set as the active path (1106).

[0130] Next, the values of each table are updated (1108). In other words, in the path table 300, the active information 303 for the path set as the active path is changed into "Yes".

[0131] Further, in the LU table 400, the value of the path ID of new active path 404 for the corresponding LU m is changed into the value of the path ID of the path set as the active path.

[0132] According to those processes, the active path for the corresponding LU m is reset.

[0133] Next, it is judged whether the active paths have been reset for all the corresponding LUs m (1109). As a result, when it is judged that the active paths have not been reset for all the corresponding LUs m (in other words, there exists an LU m for which the active path has not been reset), the procedure returns to the step 1102 to reset the active path for the subsequent LU m.

[0134] On the other hand, when it is judged that the active paths have been reset for all the corresponding LUs m, in order to put the reset active path into actual operation, the value of the path ID of new active path 404 in the LU table 400 is copied to the path ID of current active path 403 (1110).

[0135] After all those processes, the active path resetting subprogram A 207 ends its execution (1111).

[0136] FIG. 12 is a block diagram showing an example of the active paths that are reset according to the embodiment of this invention.

[0137] FIG. 12 shows an example of the active paths that are reset by the active path resetting subprogram A 207 described in FIG. 11 when there occurs a failure in the CHA 3 in FIG. 9. In this example, the active path where a failure has occurred is the path extending to the LU 3 via the HBA 1 and the CHA 3, and the measured load on the path has a value of "101".

[0138] When there occurs a failure in the CHA 3 in FIG. 9, the "corresponding LUs m" in the step 1102 is the LUs m relating to the active path including the CHA 3, in other words, the LU 3.

[0139] Therefore, the current measured loads on all the paths extending to the LU 3 are compared with each other (1102). There are six paths that extends to the LU 3, in which the path numbers shown in the path table 300 are "13" through "18". Among those, the paths (whose path numbers are "15" and "18") relating to the CHA 3 experiencing a failure cannot be used, so the measured loads on the four remaining paths are compared.

[0140] The path whose path number is "13" is the path extending to the LU 3 via the HBA 1 and the CHA 1. In the HBA table 500, the measured load on the HBA 1 has a value of "60". Here, the active path where a failure has occurred extends via the HBA 1. Thus, the value of the measured load on the active path where a failure has occurred is subtracted from the value of the measured load on the HBA 1 to obtain a value of "50". On the other hand, in the CHA table 600, the measured load on the CHA 1 has a value of "110". From the above description, the current measured load on the path whose path number is "13" is a total value of "160" obtained by summing up "50" and "110".

[0141] Similarly, the current measured loads on the path whose path numbers are "14", "16", and "17" have the values of "70", "190", and "100", respectively. By comparing those values with one another (1102), it is found that the only path where the measured load has the minimum value of "70" is the path whose path number is "14". In other words, there do not exist the plurality paths where the measured loads have the minimum value (1103), so the path whose path number is "14" is set as the active path (1107).

[0142] As a result, as shown in FIG. 12, a path 1202c extending via the HBA 1 and the CHA 2 is reset as the active path extending to the LU 3.

[0143] FIG. 13 is a flowchart of active path resetting subprogram B according to the embodiment of this invention.

[0144] In the computer system of FIG. 1, in the case where the timing for resetting the active path has been reached, the active path resetting subprogram B 208 is called in the step 710 of FIG. 7 to start its execution (1301). Here, the case where the timing for resetting the active path has been reached includes: a case where an active path resetting interval (for example, 1 hour) that is previously set has elapsed; and a case where the user of the host 101 issues the active path resetting request.

[0145] First, the active path resetting subprogram B 208 judges whether the loads on the active paths are distributed appropriately. To be specific, the minimum value and the maximum value of the measured loads on all the HBAs 105 and the CHAs 107 are compared with each other to judge whether the minimum value is 90% or more of the maximum value (1302). As a result, when the minimum value is 90% or more of the maximum value, the loads on the active paths are distributed appropriately, so the procedure advances to a step 1316 without resetting the active path.

[0146] On the other hand, when the minimum value is not 90% or more of the maximum value, there is imbalance in the loads, so the procedure advances to a step 1303 to distribute the loads by resetting the active path.

[0147] In the example of FIG. 13, the "90% of the maximum value among the measured loads" is set as a threshold based on which it is judged whether the loads on the active paths are distributed appropriately. However, the user may set another threshold (for example, 50% of the maximum value).

[0148] Next, it is judged whether or not the decision process as to whether the active path is to be changed has ended for all the LUs 109 (1303). When the decision process is judged to have ended, the procedure advances to a step 1309 to decide the active path for the LU 109 for which the active path is to be changed.

[0149] On the other hand, the decision process as to whether the active path is to be changed has not ended for all the LUs 109, the procedure advances to a step 1304 to end the process. In other words, the values of the measured loads on the LUs 109 for which it is not decided as to whether the active paths are to be changed are compared with each other, and the LU 109 (LU m, where "m" represents an LU-ID) where the value is largest is selected (1304).

[0150] Next, it is judged whether a failure has occurred in the current active path extending to the LU *m* (1305).

[0151] In the step 1305, when it is judged that a failure has not occurred in the current active path extending to the LU *m*, it is subsequently judged whether the current active path for the LU *m* is used as the new active path for another LU 109 (1306). To be specific, it is judged whether any of the HBA 105 and the CHA 107 relating to the current active path for the LU *m* is included in the new active path for another LU 109.

[0152] In the step 1306, when it is judged that none of the HBA 105 and the CHA 107 relating to the current active path for the LU *m* are included in the new active path for another LU 109, it is decided that the active path for the LU *m* is not to be changed (1308). In other words, the value of the path ID of new active path 404 corresponding to the LU *m* in the LU table 400 is set to be the same as the path ID of current active path 403. Further, in the HBA table 500 and the CHA table 600, the value of the measured load on the LU *m* (in other words, the value of the measured load 307 on the active path relating to the LU *m* in the path table 300) is added to the values of the allocated loads 503 and 603 corresponding to the LU *m*. Furthermore, in the HBA table 500 and the CHA table 600, "1" is added to the values of the active path set LU numbers 504 and 604 corresponding to the LU *m*. After that, the procedure returns to the step 1303.

[0153] On the other hand, in the step 1305, when it is judged that a failure has occurred in the current active path extending to the LU *m*, and when it is judged in the step 1306 that any of the HBA 105 and the CHA 107 relating to the current active path for the LU *m* are included in the new active path for another LU 109, it is decided that the active path for the LU *m* is to be changed (1307), and the procedure returns to the step 1303.

[0154] In the step 1303, when it is judged that the decision process as to whether the active path is to be changed has ended for all the LUs 109, it is judged whether the new active path has been decided for all the LUs 109 (1309). When it is judged that the new active path has been decided for all the LUs 109, the procedure advances to a step 1315.

[0155] On the other hand, when it is judged that the new active path has not been decided for all the LUs 109, the values of the measured loads on the LUs 109 for which new active paths are not decided are compared with each other, and the LU 109 (LU *m*) where the value is largest is selected (1310).

[0156] Next, it is judged whether there exist a plurality of paths that are usable (in other words, where no failure has occurred) for the LU *m* (1311).

[0157] When there exists only one path that is usable, the path is decided to be the new active path (1312).

[0158] On the other hand, when there exist the plurality of paths that are usable, in the HBA table 500 and the CHA table 600, the values of the allocated loads 503 and 603 corresponding to the LU *m* are referenced to decide the path where the sum of those values is minimum to the new active path (1313).

[0159] Next, the values of each table are updated (1314). In other words, with regard to the HBA 105 and CHA 107 relating to the new active path decided in the step 1312 or

1313, the value of the measured load on the LU *m* is added to the values of the allocated loads 503 and 603 in the HBA table 500 and CHA table 600. Further, "1" is added to the values of the active path set LU numbers 504 and 604. Furthermore, the value of the path ID of new active path 404 corresponding to the LU *m* in the LU table 400 is changed into the value of the path ID of new active path decided in the step 1312 or 1313. After that, the procedure returns to the step 1309.

[0160] In the step 1309, when it is judged that the new active path has been decided for all the LUs 109, the value of the path ID of new active path 404 in the LU table 400 is copied to the path ID of current active path 403 (1315). As a result, the new active path is put into actual operation.

[0161] Next, the values of the measured loads 307, 402, 502, and 602 are cleared from the path table 300, LU table 400, HBA table 500, and CHA table 600 (1316). In other words, those values are set to "0".

[0162] After all those processes, the active path resetting subprogram B 208 ends its execution (1317).

[0163] FIG. 14 is a block diagram showing another example of the active paths that are reset according to the embodiment of this invention.

[0164] FIG. 14 shows an example of the active paths that are reset by executing in FIG. 9 the active path resetting subprogram B 208 described in FIG. 13. FIGS. 4 to 6, 9, 13, and 14 will be referenced to describe the resetting procedure. It should be noted that when the same operation is repeated by a loop, the description of the repeated operation will be omitted unless it is particularly necessary.

[0165] In FIG. 9, when the active path resetting subprogram B 208 starts its execution (1301), it is first judged whether the loads on the active paths are distributed appropriately (1302). For example, in FIG. 6, a value of "99", which is 90% of the maximum value "110" of the measured load 602 on the CHA 107, is much larger than the minimum value "10". This implies that the load distribution is largely unbalanced. Therefore, the procedure advances to the step 1303 to balance the loads by resetting the active paths.

[0166] At this time, the decision process as to whether the active path is to be changed has not ended for all the LUs 109 (1303), so the procedure advances to the step 1304.

[0167] In the step 1304, the LU 4 where the measured load has the largest value is selected from among the LUs 109 for which it has not been decided whether the active path is to be changed. No failure has occurred in the current active path 902*d* extending to the LU 4 (1305), so the procedure advances to the step 1306.

[0168] Since there is no LUs 109 for which the new active path was decided before the LU 4, any of the HBA 105 and CHA 107 included in the current active path 902*d* for the LU 4 is not decided for the new active path for another LU 109 (1306). Thus, the procedure advances to the step 1308.

[0169] In the step 1308, the current active path. 902*d* for the LU 4 (path extending to the LU 4 via the HBA 2 and the CHA 1) is decided to be the new active path. At this time, the value of the path ID of new active path for the LU 4 in the LU table 400 is set to "214". Further, the value "60" (measured load on the LU 4) is added to the value "0" (initial

value) of the allocated load **503** on the HBA **2** in the HBA table **500** to obtain a value “60”. Furthermore, the value “60” (measured load on the LU **4**) is added to the value “0” (initial value) of the allocated load **603** on the CHA **1** in the CHA table **600** to obtain a value “60”.

[0170] Next, the LU **1** where the measured load has the second largest value is selected (**1304**). The CHA **1** along the current active path **902a** for the LU **1** (path extending to the LU **1** via the HBA **1** and the CHA **1**) has been decided for the new active path for the LU **4** (**1306**). Therefore, it is decided to change the active path for the LU **1** (**1307**).

[0171] It is also decided to change the active path for the LU **2** in the same manner (**1307**).

[0172] Since the current active path **902c** for the LU **3** (path extending to the LU **3** via the HBA **1** and the CHA **3**) do not overlap with the new active path **902d** for the LU **4** (path extending to the LU **4** via the HBA **2** and the CHA **1**) that has been already decided, it is decided not to change the active path (**1306** and **1308**). At this time, the value of the path ID of new active path for the LU **3** in the LU table **400** is set to “133”. Further, the value “10” (measured load on the LU **3**) is added to the value “0” (initial value) of the allocated load **503** on the HBA **1** in the HBA table **500** to obtain a value “10”. Furthermore, the value “10” (measured load on the LU **3**) is added to the value “0” (initial value) of the allocated load **603** on the CHA **3** in the CHA table **600** to obtain a value “10”.

[0173] Next, the procedure advances to the step **1309**.

[0174] At this time, since the new active paths for the LU **1** and the LU **2** are undecided, the LU **1** where the measured load has the largest value is selected (**1310**).

[0175] Since there are a plurality of paths that are usable for the LU **1** (**1311**), the procedure advances to the step **1313**.

[0176] In the step **1313**, the HBA table **500** is first referenced. At this time, the value of the allocated load **503** on the HBA **1** is “10”, and the value of the allocated load **503** on the HBA **2** is “60”. Thus, the HBA **1** where the allocated load **503** has a smaller value is decided for the new active path for the LU **1**.

[0177] Next, the CHA table **600** is referenced. At this time, the value of the allocated load **603** on the CHA **1** is “60”, the value of the allocated load **603** on the CHA **2** is “0”, and the value of the allocated load **603** on the CHA **3** is “10” (initial value). Thus, the CHA **2** where the allocated load has the smallest value is decided for the new active path for the LU **1**.

[0178] Next, in the step **1314**, the value “50” (measured load on the LU **1**) is added to the value “10” of the allocated load **503** on the HBA **1** in the HBA table **500** to obtain a value “60”. Further, the value “50” (measured load on the LU **1**) is added to the value “0” of the allocated load **603** on the CHA **2** in the CHA table **600** to obtain a value “50”.

[0179] Next, the LU **2** is selected (**1310**).

[0180] In the step **1313**, the HBA table **500** is first referenced. At this time, the value of the allocated load **503** on the HBA **1** is “60”, and the value of the allocated load **503** on the HBA **2** is “60”. Thus, when the values of the allocated loads **503** are the same, either of the HBAs **1** and **2** may be

decided for the new active path. In this embodiment, the HBA **1** whose HBA-ID has a smaller value is decided for the new active path for the LU **2**.

[0181] Next, the CHA table **600** is referenced. At this time, the value of the allocated load **603** on the CHA **1** is “60”, the value of the allocated load **603** on the CHA **2** is “50”, and the value of the allocated load **603** on the CHA **3** is “10”. Thus, the CHA **3** where the allocated load **603** has the smallest value is decided for the new active path for the LU **2**.

[0182] Next, in the step **1314**, the value “20” (measured load on the LU **2**) is added to the value “60” of the allocated load **503** on the HBA **1** in the HBA table **500** to obtain a value “80”. Further, the value “20” (measured load on the LU **2**) is added to the value “10” of the allocated load **603** on the CHA **3** in the CHA table **600** to obtain a value “30”.

[0183] As described above, the new active path for each LU **109** is decided. Finally, as shown in **FIG. 14**, a new active path **1402a** extending to the LU **1** passes the HBA **1** and the CHA **2**. A new active path **1402b** extending to the LU **2** passes the HBA **1** and the CHA **3**. The new active path **902c** extending to the LU **3** passes the HBA **1** and the CHA **3**. The new active path **902b** extending to the LU **4** passes the HBA **2** and the CHA **1**.

[0184] At this time, the maximum value of the allocated load **603** on the CHA is “60”, and the minimum value thereof is “30”. On the other hand, the maximum value of the measured load **602** on the CHA relating to the current active path is “110”, and the minimum value thereof is “10”. Accordingly, it is understood that the unbalanced loads are alleviated.

[0185] According to the above embodiment of this invention, the active paths are set by designating the HBAs **105** and the CHAs **107** so as to balance the loads on the HBAs **105** and the CHAs **107**. However, the active paths may be set by designating any one of the HBAs **105** and the CHAs **107** so as to balance the loads on the designated one of the HBAs **105** and the CHAs **107**. For example, weighting may allow the user to select which one of the loads on the HBAs **105** or the CHAs **107** is balanced prior to the other.

[0186] For example, in the case of setting the active paths by designating only the CHA **107**, in the step **809** of **FIG. 8**, only the values of active path set LU numbers **604** in the CHA table **600** are compared with one another to set the initial active paths. Further, in the steps **1102** and **1103** of **FIG. 11**, only the measured loads **602** in the CHA table **600** are compared with one another to reset the active paths. Furthermore, in the step **1313** of **FIG. 13**, only the allocated loads **603** in the CHA table **600** are compared with one another to reset the active paths.

[0187] The active path thus set (or reset) designates only the CHA **107** without designating the HBA **105**. As a result, for example, when the active path extending to the LU **1** is set for the CHA **1**, the host **101** makes an access (I/O) to the LU **1** via an arbitrary HBA **105** and the designated CHA **1**. In other words, a plurality of paths that pass the CHA **1** are set as the active paths for the LU **1**. In other words, the host **101** always uses the CHA **1** to access the LU **1**. On the other hand, the host **101** can use different HBAs **105** on an access (I/O) basis. For example, the host **101** may adopt the conventional load distribution for the HBAs **105**.

[0188] Even when the active path is thus set, as long as the CHA 107 includes the cache memory, the cache memory is effectively used, thereby improving the performance according to the data pre-reading function or the like.

[0189] Similarly, when the HBA 105 includes the cache memory, the active paths may be set by designating only the HBAs 105. As a result, the cache memory of the HBA 105 is effectively used, thereby improving the performance.

[0190] According to this invention described above, the active path is set for each of the LUs 109, and the host 101 accesses the LUs 109 via the active paths. The active paths are set so as to balance the access loads from the host 101. As a result, the performance can be prevented from decreasing due to accesses concentrated on a specific path.

[0191] Further, after the active paths are set (or reset), and before being reset, the host 101 uses the same path to access one LU 109. In other words, since the active path is not changed for each access, the cache memory of the CHA 107 or the like included in the active path is effectively used. In particular, in sequential accesses, the pre-read data stored in the cache memory is effectively used. As a result, the performance is improved.

[0192] Further, when there is an imbalance in the access loads, the active paths are reset so as to resolve the imbalance, thereby maintaining the satisfactory performance at all times.

[0193] Further, in the case of resetting the active paths, the resetting is performed from the LU 109 where the load is large in order, thereby distributing the loads appropriately.

What is claimed is:

1. A computer system, comprising:
 - a computer; and
 - a storage system coupled with the computer, the storage system comprising:
 - one or more channel adapters which communicate with the computer; and
 - a plurality of logical units which store data, the computer comprising:
 - one or more host bus adapters which communicate with the storage system; and
 - a path management module which manages a path along which the computer accesses the logical unit,
- the path management module comprising:
 - a load measurement module which measures access loads on each logical unit; and
 - an active path setting module which selects one of: one channel adapter and one or more host bus adapters; and one or more channel adapters and one host bus adapter, along which an access from the computer passes, for each of the logical unit based on the measured access loads, and sets an active path that passes the selected channel adapter and the selected host bus adapter.
2. The computer system according to claim 1, wherein the active path setting module:
 - sums up the measured access loads on the basis of at least one of the channel adapter and the host bus adapter; and

- sets an active path that passes at least one of the channel adapter and the host bus adapter in each of which a total of the access loads is smallest.

3. The computer system according to claim 2, wherein the active path setting module:

- when there are plural channel adapters where a total of the access loads is smallest,

- calculate the number of set active paths for each of the channel adapter; and

- sets an active path that passes the channel adapter where the total of the access loads is smallest and the calculated number of active paths is smallest.

4. The computer system according to claim 2, wherein the active path setting module:

- when there are plural host bus adapters where a total of the access loads is smallest,

- calculate the number of set active paths for each of the host bus adapter; and

- sets an active path that passes the host bus adapter where the total of the access loads is smallest and the calculated number of active paths is smallest.

5. The computer system according to claim 2, wherein the active path setting module:

- compares the measured access loads on each logical unit; and

- sets the active path for the logical unit where the measured access load is larger prior to the active path for the logical unit where the measured access load is smaller.

6. The computer system according to claim 1, wherein the measured access load is represented by the number of accesses measured per predetermined time.

7. The computer system according to claim 1, wherein the measured access load is represented by an access data amount measured per predetermined time.

8. A program which is executed by a computer coupled with a storage system,

- the storage system comprising: one or more channel adapters which communicate with the computer; and a plurality of logical units which store data, the computer comprising one or more host bus adapters which communicate with the storage system,

- the program comprising:

- a first procedure of measuring access loads on each logical unit; and

- a second procedure of selecting one channel adapter and one or more host bus adapters, along which an access from the computer passes, for each of the logical unit based on the measured access loads, and setting an active path that passes the selected channel adapter and the selected host bus adapter.

9. The program according to claim 8, wherein the second procedure comprises the procedures of:

- summing up the measured access loads on the basis of the at least one of the channel adapter and the host bus adapter;

- in a case where at least one of the channel adapter and the host bus adapter where a total of the access loads is

smallest exist for a single number per adapter, setting an active path that passes at least one of the channel adapter and the host bus adapter where the total of the access loads is smallest; and

in a case where the at least one of the channel adapter and the host bus adapter where a total of the access loads is smallest exist for a plural number, calculating the number of set active paths for each of the at least one of the channel adapter and the host bus adapter, and setting an active path that passes at least one of the channel adapter and the host bus adapter where the total of the access loads is smallest and the calculated number of active paths is smallest.

10. A method for setting an active path along which a computer accesses a logical unit in a computer system which comprises:

the computer; and

a storage system coupled with the computer,

the storage system comprising: one or more channel adapters which communicate with the computer; and a plurality of logical units which store data,

the computer comprising one or more host bus adapters which communicate with the storage system,

the method comprising:

a first step of measuring access loads on each logical unit; and

a second step of selecting one channel adapter and one or more host bus adapters, along which an access from the computer passes, for each of the logical unit based on the measured access loads, and setting an active path that passes the selected channel adapter and the selected host bus adapter.

11. The method according to claim 10, wherein the second step comprises the steps of:

summing up the measured access loads on the basis of the at least one of the channel adapter and the host bus adapter;

in a case where at least one of the channel adapter and the host bus adapter where a total of the access loads is smallest exist for a single number per adapter, setting an active path that passes at least one of the channel adapter and the host bus adapter where the total of the access loads is smallest; and

in a case where the at least one of the channel adapter and the host bus adapter where a total of the access loads is smallest exist for a plural number, calculating the number of set active paths for each of the at least one of the channel adapter and the host bus adapter, and setting an active path that passes at least one of the channel adapter and the host bus adapter where the total of the access loads is smallest and the calculated number of active paths is smallest.

* * * * *